# Fișă de lucru

# Pb 1.

Să se realizeze o interfaţă pentru gestionarea persoanelor din evidenţa unei asociaţii de locatari. Se utilizează o bază de date cu o tabela având următoarea structură: ID(N,6), NUME(C,20),NR\_BLOC (C,5), NR\_AP(N,4), ANGAJAT(L). Să se creeze baza de date şi tabela, introducând 10 înregistrări. Interfaţa trebuie să ofere posibilitatea efectuării următoarelor acţiuni:

1. Introducerea de noi persoane în evidenţă,
2. Stergerea unei persoane dupa ID
3. Afişarea tuturor persoanelor care sunt angajate.
4. Afisarea locatarilor unui bloc dat
5. Afisarea tuturor persoanelor ordonate dupa numarul blocului, apoi dupa nume alfabetic

Rezolvare în **C#:**

1. Se crează o aplicaţie de tip **WindowsFormsApplication** numită **AsociatieLocatari**.
2. Pentru a crea baza de date, click dreapta pe numele proiectului, **Add →** **New** I**tem** şi alegeţi **Service-based Database**, daţi numele **asociaţie.mdf** şi apăsaţi butonul **Add**. Aşteptaţi puţin până apare fereastra **Data Source Configuration Wizard**, se va crea **DataSet**-ul cu numele implicit **asociatieDataSet**. Apăsaţi butonul **Finish**.
3. În tab-ul **Database Explorer** **(**apare de obicei unde este şi Toolbox-ul. Dacă nu apare, alegeţi din meniul **View**, opţiunea **Other Windows** şi apoi alegeţi **Database Explorer)**, expandaţi nodul **asociaţie.mdf**, daţi **click dreapta pe Tables** şi alegeţi **Add New Table** pentru a adăuga o nouă tabelă în baza de date. Completaţi structura tabelei după cerinţele problemei. Veţi alege tipurile câmpurilor astfel:

* pentru **ID** – alegeţi tipul **numeric(6,0)** şi din fereastra **Column Properties** alegeţi pentru proprietatea **Identity Specification** valoarea **Yes**, cu **Identity Increment** având valoarea **1** şi **Identity Seed** tot valoarea **1** (astfel calculatorul va completa automat valorile din câmpul ID începând de la 1, valori consecutive, nepermiţând ca în acest câmp să fie două valori identice)
* pentru **NUME** – alegeţi tipul nvarchar, iar din fereastra **Column Properties** alegeţi pentru proprietatea **Length** valoarea **20**
* pentru **NR\_BLOC** – alegeţi tipul **nvarchar**, de lungime **5**.
* pentru **NR\_AP** – alegeţi tipul **int**.
* pentru **ANGAJAT** – alegeţi tipul **bit**, care permite ca valori doar: **true** şi **false**.

Toate campurile trebuie completate obligatoriu, nu se permite valoarea Null.

1. Salvaţi tabela cu numele **locatari.**
2. În fereastra **Database Explorer**, daţi **click dreapta pe** tabela **locatari** şi alegeţi opţiunea **Show Table Data.** Adăugaţi 10 înregistrări.
3. Pe forma **Form1**, adăugaţi din **Toolbox** un control de tip **MenuStrip** şi completaţi cu opţiunile: **Operatii** ( care are in submeniu optiunile **Adauga persoana**  si **Sterge persoana**)***, Afisari*** *(**care are in submeniu optiunile* ***Doar cei angajati, Locatarii unui bloc*** *si* ***Toti ordonati dupa bloc, apoi dupa nume alfabetic)*** şi ***Ieşire***.
4. Din meniul **Data**, selectaţi opţiunea **Show Data Sources**. În fereastra **Data Sources**, daţi **click dreapta pe** elementul **asociaţieDataSet**, apoi selectaţi **Configure DataSet with Wizard…** În dialogul **Data Source Configuration Wizard**, selectaţi tabela **locatari**, apoi apăsaţi butonul **Finish**.
5. În fereastra **Data Source**, selectaţi tabela **locatari** şi trageţi-o pur şi simplu pe **Form1**. Va apărea automat un control de tip **GridView** în care vor putea fi afişate informaţiile din tabelă. Stergeti de pe forma l**ocatariBindingNavigator**.
6. Pentru verificare apăsaţi tasta **F5** sau butonul **Start Debugging**.
7. Pentru a introduce în evidenţă o nouă persoană, se va afişa o nouă formă în a cărei controale vom introduce informaţiile. Aceste controale vor fi legate la tabela **locatari**. Forma va apărea pe ecran în momentul în care vom da click în meniu pe opţiunea ***Adauga persoana***. Pentru aceasta parcurgeţi paşii următori:

* Pentru a adăuga priectului o nouă formă, în fereastra **Solution Explorer**, daţi c*lick dreapta pe numele proiectului*, selectaţi **Add** şi apoi **Windows Form**. Lăsaţi în acest exemplu numele implicit, **Form2**.
* În fereastra **Data Source**, selectaţi tabela **locatari**, deschideţi lista ascunsă din dreptul ei şi selectaţi opţiunea **Details**. Selectaţi apoi tabela şi trageţi-o pur şi simplu pe **Form2**. Vor apărea automat controale de tip textBox şi checkBox (legate la tabela locatari) în care vor putea fi introduse informaţii noi. Pentru controlul **checkBox1** setaţi proprietatea **Text** la şirul vid, iar **textBox**-ul pentru **ID** si **label**-ul corespunzator stergeti-le deoarece aceasta informatie se completeaza automat de program..
* Reveniţi în **Form1** şi pentru a trata evenimentul **Click** al opţiunii ***Adauga persoana***, veţi scrie codul următor:

**Form2 f = new Form2();**

**f.ShowDialog*();*** *// afiseaza Form2, ca dialog modal acest tip de*

*// fereastra dialog nu permite posibilitatea accesarii altor //ferestre ale aplicatiei pana cand acest dialog nu este inchis*

**locatariTableAdapter.Fill(asociatieDataSet.locatari);**

*// se vor reactualiza datele in tabela si implicit si* *in GridView*

*//* *care e legat de aceasta*

1. Pentru a afla toate persoanele *angajate* a căror informaţii sunt reţinute în tabela **locatari**, se va realiza o ***interogare***. Pentru aceasta procedaţi în felul următor:

* În fereastra **Data Sources**, daţi **click dreapta pe** tabela **locatari** şi alegeţi opţiunea **Edit DataSet with Designer.**
* Daţi **click dreapta pe locatariTableAdapter** şi alegeţi opţiunea **Add → Query.** Va apărea fereastra TableAdapter Query Configuration Wizard, în care este selectată implicit opţiunea **Use SQL statements**. Apăsaţi butonul **Next**.
* Va trebui să alegeţi ce comandă SQL vreţi să folosiţi. Va fi selectată implicit comanda care ne trebuie: **SELECT wich returns rows**. Apăsaţi butonul **Next**.
* Se va afişa implicit o comandă **SELECT** care întoarce toate înregistrările din tabela **locatari**. Aceasta va trebui modificată. Pentru aceasta, apăsaţi butonul **Query Builder…** şi modificaţi comanda ca mai jos:

**SELECT ID, NUME, NR\_BLOC, NR\_AP, ANGAJAT**

**FROM locatari**

**WHERE (ANGAJAT = 'true')**

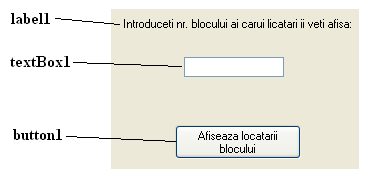
apoi apăsaţi butonul **Execute Query** pentru a verifica corectitudinea comenzii. Dacă comanda e bună, apăsaţi butonul **OK,** apăsaţi butonul **Next**. Observaţi că automat vi s-au generat metodele **FillBy** şi **GetDataBy** cu ajutorul cărora veţi face legătura dintre interogarea voastră şi controalele în care afişaţi rezultatul interogării. Redenumiti-le numele **FillByAngajati**, respectiv **GetDataByAngajati** .

* Apăsaţi butonul **Next** şi apoi **Finish** pentru a termina interogarea.
* Pe **Form1**, daţi dublu click în **MenuStrip** pe opţiunea **Afisari** apoi pe ***Doar cei angajati*** pentru a trata evenimentul **Click** şi scrieţi codul următor pentru a afişa în controlul de tip GridView doar persoanele angajate. Veţi afişa de fapt rezultatul interogării realizate anterior.

**locatariTableAdapter.FillByAngajati(asociatieDataSet.locatari);**

Pentru a afisa locatarii unui bloc vom proceda astfel:

Adăugaţi în proiect o nouă formă denumită **Form3**. Pe aceasta adăugaţi din **ToolBox** trei controale ca în figură:



1. În fereastra **Data Source**, selectaţi tabela **locatari**, deschideţi lista ascunsă din dreptul ei şi selectaţi opţiunea **DataGridView**. Selectaţi apoi tabela şi trageţi-o pur şi simplu pe **Form3**. Va apărea automat un control de tip **GridView** în care vor putea fi afişate informaţiile filtrate. . Stergeti de pe forma l**ocatariBindingNavigator**.

În clasa **Form3** adăugaţi o dată membră publică de tip *string* numită **bloc** astfel:

**public string bloc;**

În continuare realizaţi pe rând următorii paşi (aceeaşi ca la crearea interogării anterioare doar că se va schimba comanda şi numele metodelor generate la final):

* În fereastra **Data Sources**, daţi **click dreapta pe** tabela **locatari** şi alegeţi opţiunea **Edit DataSet with Designer.**
* Daţi **click dreapta pe locatariTableAdapter** şi alegeţi opţiunea **Add → Query.** Va apărea fereastra TableAdapter Query Configuration Wizard, în care este selectată implicit opţiunea **Use SQL statements**. Apăsaţi butonul **Next**.
* Va trebui să alegeţi ce comandă SQL vreţi să folosiţi. Va fi selectată implicit comanda care ne trebuie: **SELECT wich returns rows**. Apăsaţi butonul **Next**.
* Se va afişa implicit o comandă **SELECT** care întoarce toate înregistrările din tabela **locatari**. Aceasta va trebui modificată. Pentru aceasta, apăsaţi butonul **Query Builder…** şi modificaţi comanda ca mai jos:

**SELECT ID, NUME, , NR\_AP, ANGAJAT**

**FROM locatari**

**WHERE ( NR\_BLOC = @bloc )**

apoi apăsaţi butonul **Execute Query** pentru a verifica corectitudinea comenzii.

***Observaţi că pentru a folosi valoarea unei date membre a unei forme se scrie în faţa ei @ (, iar data membră folosită, trebuie să fie publică) .***

Dacă comanda a fost verificată şi este bună, apăsaţi butonul **OK,** apăsaţi butonul **Next**. Observaţi că automat vi s-au generat metodele **FillBy** şi **GetDataBy.** Redenumiţi-le cunumele sugestive: **FillByBloc,** respectiv **GetDataByBloc** .

* Apăsaţi butonul **Next** şi apoi **Finish** pentru a termina interogarea.
* Pe **Form1**, daţi dublu click în **MenuStrip** pe opţiunea ***Afisari,*** *apoi* ***Locatarii unui bloc***  *si pentru* evenimentul **Click** şi scrieţi codul următor pentru a afişa **Form3** (în care veţi introduce numărul blocului şi veţi a afişa într-un control de tip GridView doar persoanele din blocul dat. Veţi afişa de fapt rezultatul interogării realizate anterior)

**Form3 f3 = new Form3();**

**f3.ShowDialog();**

În **Form3**, în handler-ul de tratare a evenimentului **Click** pentru **button1** scrieţi următorul cod care salvează valoarea introdusă în **textBox1** în data membră **bloc** şi care afişează rezultatul interogării realizate anterior:

**bloc = textBox1.Text;**

**locatariTableAdapter.FillByBloc(asociatieDataSet.locatari,bloc);**

În continuare realizaţi pe rând următorii paşi (aceeaşi ca la crearea interogării anterioare doar că se va schimba comanda şi numele metodelor generate la final) pentru a afisa toti locatarii ordonati dupa bloc si apoi dupa nume alfabetic:

* Creati un **Query** ca mai jos:

**SELECT ID, NUME, NR\_BLOC, NR\_AP, ANGAJAT**

**FROM locatari**

**ORDER BY NR\_BLOC, NUME**

Numele metodelor generate vor fi FillByOrdonati si GetDataByOrdonati.

* Pe **Form1**, daţi dublu click în **MenuStrip** pe opţiunea ***Afisari,*** *apoi* ***Toti ordonati dupa bloc, apoi dupa nume alfabetic***  *si pentru* evenimentul **Click** şi scrieţi codul următor:

**locatariTableAdapter.FillByOrdonati(asociatieDataSet.locatari);**

Pentru a sterge o persoana dupa ID, vom proceda astfel:

* **Pe Form1,** sub controlul de tip **DataGridView** vom adauga din ToolBox un mic Panel, pe care vom plasa un coltrol de tip label care va avea implicit numele **Label1** si vom seta textul la *ID*, un textBox care va avea numele implicit **textBox1** si un buton cu numele **buton1** si care va avea textul *Sterge persoana*.
* În fereastra **Data Sources**, daţi **click dreapta pe** tabela **locatari** şi alegeţi opţiunea **Edit DataSet with Designer.**
* Daţi **click dreapta pe locatariTableAdapter** şi alegeţi opţiunea **Add → Query.** Va apărea fereastra TableAdapter Query Configuration Wizard, în care este selectată implicit opţiunea **Use SQL statements**. Apăsaţi butonul **Next**.
* Va trebui să alegeţi ce comandă SQL vreţi să folosiţi. Selectati **DELETE**. Apăsaţi butonul **Next**.
* Modificaţi comanda DELETE ca mai jos:

**DELETE FROM locatari**

**WHERE (ID = @id)**

apăsaţi butonul **Next**. Observaţi că automat s-a generat metoda **DeleteQuery**. Redenumiti-o cu numele **DeleteQueryID** .Apăsaţi butonul **Next** şi apoi **Finish** pentru a termina interogarea.

* **Pe Form1,** in panel dati dublu click pe butonul **Sterge persoana** si scrieti codul:

**locatariTableAdapter.DeleteQueryID(int.Parse(textBox1.Text));**

**locatariTableAdapter.Fill(asociatieDataSet.locatari);**

Scrieti cod care va face panelul initial invizibil si va deveni vizibil doar la alegerea optiunii **Sterge persoana** din meniul  **Operatii.**

**Rulaţi şi testaţi aplicaţia.**

# Pb. 2.

Să se realizeze o interfaţă pentru gestionarea cărţilor dintr-o bibliotecă. Se utilizează o bază de date cu o tabelă având următoarea structură: ISBN(C,14), AUTORUL(C,20), TITLUL(C,25), ANUL\_AP(N,4), NR\_VOL(N,4). Să se creeze baza de date şi tabela, introducând 10 înregistrări. Interfaţa trebuie să ofere posibilitatea efectuării următoarelor acţiuni:

1. Introducerea de noi cărţi în evidenţă,
2. Afişarea cărţilor care sunt într-un număr de *n* volume, *n* fiind specificat.

# Pb. 3.

Să se realizeze o interfaţă pentru gestionarea produselor dintr-o magazie. Se utilizează o bază de date cu o tabelă având următoarea structură: COD(N,6), DENUMIREA(C,30), PRET(N,8.2), DATA\_IN(D). Să se creeze baza de date şi tabela, introducând 10 înregistrări. Interfaţa trebuie să ofere posibilitatea efectuării următoarelor acţiuni:

1. Introducerea de noi produse,
2. Afişarea produselor care au preţurile cuprinse între două limite specificate.

# Pb. 4.

Să se realizeze o interfaţă pentru gestionarea zborurilor dintr-un aeroport. Se utilizează o bază de date cu o tabelă având următoarea structură: ID(N,4), DESTINATIA(C,25), DURATA(N,3), LOC\_LIB(N,3), DATA\_PLE(D). Să se creeze baza de date şi tabela, introducând 10 înregistrări. Interfaţa trebuie să ofere posibilitatea efectuării următoarelor acţiuni:

1. Introducerea de noi zboruri,
2. Anularea unui zbor,
3. Listarea locurilor libere, spre orice destinaţie, pentru o data specificată.